DESIGN OF 6 DOF ROBOTIC ARM CONTROLLED OVER THE INTERNET

[1]Rajiv. G, [2]Pradeep. R,
[1,2]Student, 1Sathyabama University, [2]Saveetha Engineering College, Chennai, INDIA
Rajiv.srkm@gmail.com, Pradeep334@live.com

Abstract— The purpose of the project is to build a robotic arm, which can be controlled remotely over the internet, and to perform operations exhibiting higher accuracy with 6 degrees of freedom. The robotic arm uses IP network to receive data/commands from the user, over the internet, which are interpreted into proper instructions and then relayed to the controller. It reduces the human effort when used for applications such as nuclear waste disposal and bomb disposal when compared to present methods.

Index Terms — *Microcontroller ATMEGA32, Robotic Arm, UDP protocol, Internet*

1. INTRODUCTION

A robotic arm designed using motors is a mechanical arm, which can be autonomous or remotely controlled, having multipurpose manipulator programmable in three or more axes and can be used to perform a variety of tasks with great accuracy and speed.

The ROBOTIC ARM with six Degrees Of Freedom (DOF) similar to a human arm can perform all tasks (including the tasks performed by a human) with ease and comparatively faster, simpler with fewer movements. In the present scenario Robotic arm has fewer degrees of freedom typically less than five, but for a successful replication of a human ARM we require 6 DOF. The replication of a human arm would be ideal to use it as a bionic arm or to perform remotely controlled industrial automations in a simpler and accurate manner.

In our project we have designed a Robotic ARM with servo motors [3]. The selection of the servo motors was due to their ability to move to a finer angle, High Torque during loaded and unloaded conditions versus DC motors and stepper motors have low holding torque with less precision. These attributes made the servo motor an effective choice for the construction of the ARM. In the project a user interface (GUI) (front end GUI) was created using VB.NET which is used to control the ARM. The GUI contains a slider control for the control of the servo motors. On moving the slider in the windows form, corresponding signals are sent in order to move the servo motors. The VB.NET interface is responsible for sending the data to the microcontroller (MC) over internet. The data is at first encoded and then start and stop bits are added with check bits to ensure the reliability of data. Then the data is received by the microcontroller, where it is decoded and checked for reliability of transmission using check bits, start and stop bits.

Due to the requirement of several PWM channels, The Atmega32 Micro controller was used to resolve our limitations [5]. The flexibility of the Atmega32 alongside with a number of programmable ports with multiple PWM channels was an ideal choice for end implementation.

The decoded information is converted into digital data and is updated along with the address. The Atmega32 reads the data and then, generates the PWM and updates the value to the corresponding joint motor. The microcontroller then checks for the PWM pulses by taking an internal feedback from the generated data's and then checks the values with pre defined values. It then sends the feedback (ACK packet) to the MC which then forwards it back to the GUI (Server) for successful validation. The robot also features isolators which are used to isolate the MC and the motor drivers to avoid reverse voltage protection to other circuits. The signal from the isolation circuits are amplified to the desired voltage level and sent to the motors.

The entire ARM is constructed on a ROBOT which can be moved via positioning control through GUI. The focus of this work is effective use of arm control through computer networks, based on IP protocol. The GUI runs on server computer and it receives control data from client computer (GUI Client) and operates according to the received data.

The ROBOT functions based on two modes of operation

- Remote Mode - In this mode the ROBOT can be operated via Wi-Fi or through Internet
- Auto Mode - In this mode the ROBOT and the ARM can be pre-programmed to perform certain task or tasks in a sequential manner without any human intervention.

.

In the Auto mode, the ROBOT features a camera, which is used for image processing, for visual feedback and motion control. The presence of camera helps the ROBOT to find an approximate estimate of distance between objects. It also helps in determining the distance from the ROBOT to the object. It can be re-programmed to suite user applications.

2. MICROCONTROLLER

2.1 *Overview*: A microcontroller is a small computer on single integrated circuit chip containing a processor core, memory and programmable I/O peripherals. Program memory is in the form of flash or OTP ROM is also included on chip. Microcontrollers are designed for embedded applications and it delivers high performance at low cost for various applications.

2.1.1 *ATmega32:* Microcontroller used in the Robot for processing and control of the DATA and ADDRESS packets was Atmega32. The Atmega32 is an AVR family micro controller produced by ATMEL. They have considerable processing power and enough memory and are easy to program [5]. It has separate buses for program and data memory for faster access. The features of Atmega32 were ideal for the intended purpose. The micro controller communicates with the user interface via serial port, with a baud rate of 9600bps. The micro controller is responsible for the communication between the user GUI Server and the servo motors.
The MC on receiving the packet from the GUI Server splits the packet into header, data, parity, footer. It uses the start and end fields to recognize the start and end of the transmission [1]. Then the header field information is used to determine whether the received data is either ADDRESS or PWM DATA. Then the parity field and footer field is used to verify the correctness of data. On successful validation an ACK packet is sent to the GUI. The fields of the ACK packet contain.

START	ACK	END

START –Indicates the start of transmission.
ACK –Contains ACK of the micro controller
END –Indicates end of transmission.

When the data cannot be verified, a NACK packet is sent. The NACK packet contains

START	NACK	END

START - Indicates the start of transmission.
NACK - Contains NACK of the micro controller
END - Indicates the end of transmission.
The MC after successful validation of the packet, sent by UI, sends the PWM DATA through a parallel bus. It then waits for an ACK signal from motor controller. On successful reception on ACK signal it updates the Bus with the data value. If updated it sends an ACK signal back or else a NACK signal is sent.
On receiving a NACK signal, the microcontroller retransmits the data via bus for 5 trials. On failure to obtain an ACK signal after 5 trials, will lead to the transmission of a NACK packet from the MC to the GUI Server which is relayed to the GUI Client. On successful reception of the ACK signal from motor controller an ACK packet is sent. The feedback ensures the implementation of the PWM signal, and also helps to debug the problem in a simple manner.

2.2. *PWM:* Pulse-width modulation (PWM), as it applies to motor control, is a way of delivering energy through a succession of pulses rather than a continuously varying (analog) signal. By increasing or decreasing pulse width, the controller regulates energy flow to the motor shaft.
In our case the servo motors which should be made to rotate according to desired angle is controlled using PWM signal. PWM signal used for controlling servo motors has constant pulse width of 20ms (50 Hz) whose ON time is varied according to desired position. For the entire servo motors pwm width is constant. Initially motor shaft is adjusted to 90 degrees and forward and reverse movements are controlled by slider application in visual basic.

Duty cycle = T_{on} / Time period

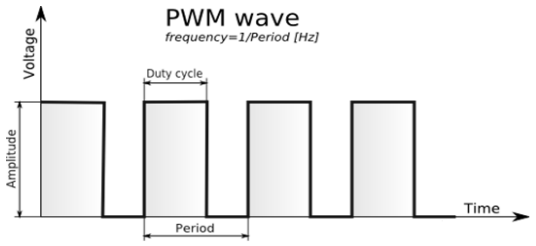

Fig 2.1 PWM Waveform

Following are ON time associated with degree of rotation,
ON time for **zero** degree position of shaft: 0.7ms (3.5% DC)
ON time for 180 degree position of shaft: 2.3ms (11.5%DC)
The above range (0.7ms to 2.3ms) is divided by 256 (8 bit output) outputs from slider.
Initial position (downwards) = 0 (00000000) =0.7ms
Intermediate position (middle) = 128 (10000000) =1.5ms
Final position of slider (upwards) = 255(11111111) =2.3ms
Therefore,
(2.3-0.7)*10^-3/ 256 =6.25us
So for 1 degree movement, ON time should be incremented by 6.25us

Degree of rotation (deg)	**ON time (ms)**
0	0.70000
1	0.70625
2	0.71250
3	0.71875
.	.
.	.
90	1.5000
.	.
.	.

180	2.3000

Table 2.1 ON Time for each Degrees of rotation

The various PWM signals are generated for each individual motors. The PWM signals will have various ON and OFF times mentioned in the Table 2.1. Hence for each degree a corresponding PWM signal is generated.

2.3. *Operations in Atmega32:*

The Atmega32 MC receives the data and address through the bus. The MC sends an ACK signal for each received ADDRESS and it also sends an ACK signal after each successful implementation of PWM DATA received for the motor. It sends an NACK after failure of implementation. The MC uses the PWM DATA and ADDRESS to update the PWM value generated for that address. For instance, a PWM value of 255 on updating channel 10 (Previous PWM value 0) would result in a 180 degree turn by the Servo motor on channel 10. Thus to change the angle of the Servo motor, we need to change the PWM value in the DATA field.

2.4. *Microcontroller Connection diagram:*

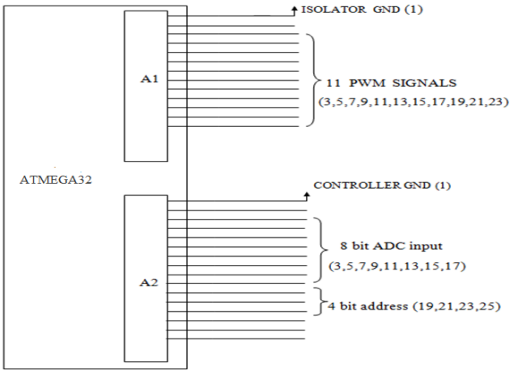

Fig 2.2. Connection diagram with Atmega32

3. USER INTERFACE WITH ARM

The Windows application in Microsoft Visual studio was selected for the GUI due to its ability to easily support Input/output operations via serial port (RS232) and its ease of use to establish a client and Server architecture. In the remote mode the GUI client accepts user instructions via slider and transmits them over internet to GUI Server which then converted into suitable data. The GUI can be accessed through another computer over internet. To ensure reliable transmission the packet transferred contains the following attributes.

START	HEADER	DATA	PARITY	FOOTER	END

START- Indicates the start of transmission.

HEADER - Contains information regarding PWM Data or Address.
DATA- Contains the PWM DATA or ADDRESS.
PARITY- It contains the parity element.
FOOTER- Contains the length of the packet.
END- Indicates the end of transmission.

To move a motor to 180 degrees, a data of 255 must be sent from the computer to the MC in order to generate the PWM signals. Each motor is assigned with a specific ADDRESS. For example if a motor has its ADDRESS as '6'. The header information of ADDRESS "6" is added with the data 255. Now the parity bit is added with the data. Then the length of the data is calculated and the footer information is added in the footer field. Then the START and END fields is added to the packet.
The microcontroller on receiving the packet splits the individual fields and determines the data and checks for parity and verifies the length of the data and the correctness of Data [4]. On successful verification an Acknowledgment Packet (feedback) is sent to the User Interface. The User interface then waits for the Acknowledgment packet. If the microcontroller receives the data and generates the exact PWM signal then an ACK packet is sent to the GUI for verification. Thus the validation of data and its successful implementation is performed and the result is updated in the GUI.
On failure of receiving the data, a NACK packet is sent by the corresponding device (MC) to the GUI. On receiving such a packet the GUI attempts to send the ADDRESS and data again for a sequence of 5 attempts after which it reports an error to the End User for probable connection problems.

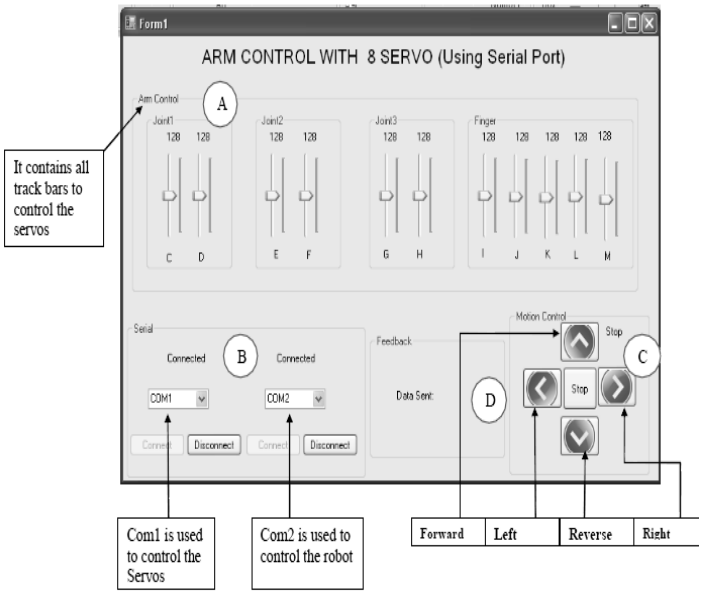

Fig 3.1 Window form application for GUI interface

4. CONTROL OVER INTERNET

The structure of Control over internet consist of two independent computer connected to the internet. One is the controller (GUI Client) and other is robotic arm on-board computer. The arm receives the control signal from the controller and applies them to the microcontroller. This control is very useful when the user doesn't have direct access to arm.

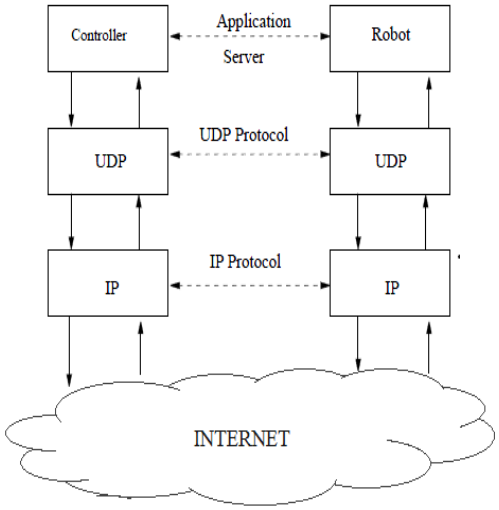

Fig 4.1 Layers of control network

The transport layer shown in figure 4.1 uses UDP protocol instead of the most used TCP protocol. The TCP protocol is connection oriented and uses error detections and correction method in order to provide a reliable connection between source and destination. The error recovery is based on retransmission of lost packets [2]. The UDP protocol is a datagram oriented protocol. Messages are transmitted in the hope that they arrive at destination but there is not any verification that they actually reached their destination. In our project TCP doesn't make any commitment with respect to message timing. In control network, the messages with large delay of no use, since the control signal they carry are out-dated. Messages discarded by application protocol due to large delay are also harmful since they used portion of system bandwidth which potentially cause delay to other messages. In this context message retransmission as performed TCP tends to be also harmful since retransmission also suffers large delay than regular messages. In extreme case, retransmission cause network congestion which forces messages to be discarded[3].

On the other hand, in a digital control system, usually sensors are sampled at constant rate. Therefore messages with sensor readings are sampled at constant rate. For the control of robotic arm whenever a message is received with errors, it could be more convenient to wait for the next message with sensor readings than retransmit the lost message to receive out-dated readings. Also prior sensor readings can be used to estimate the readings on the lost messages. With these considerations it becomes natural to use UDP since it uses less bandwidth and is more efficient with respect to timing than TCP protocol. The application layer dealt with problem of delays imposed to messages.

5. ISOLATION CIRCUIT

5.1. *Overview*: The Isolation Circuit is essential for isolation between microcontroller and the Servo Motors. Since Atmega32 works at 5v it cannot be used to drive the motors as it would cause reverse current, which consequently would damage the MC [4]. Hence Isolation circuit is important to completely isolate them from each other. We used an 814A opto coupler IC for the Isolation Circuit. The Isolation circuit also provides the necessary amplification to drive servo motors. Since the voltage produced by the MC was in the order of 4.8V and the voltage required for servo operation was 8.0V the Isolation circuit provided the necessary amplification. Based on the output from the MC the circuit maintains the digital logic state of '1' or '0' for corresponding generation of PWM signal for the servo.0

5.2 Signal conditioning Circuit: A signal conditioning circuit is an electrical regulator designed to automatically maintain a constant voltage level. In our Design the signal conditioning circuit is used for connecting the Atmega32 and the computer port for bidirectional transfer of DATA, ADDRESS and control packets such as ACK, NACK. In order to avoid malfunctioning of the MC due to excessive voltage conditioning circuit is used.

5.3. Isolation and conditioning circuits:

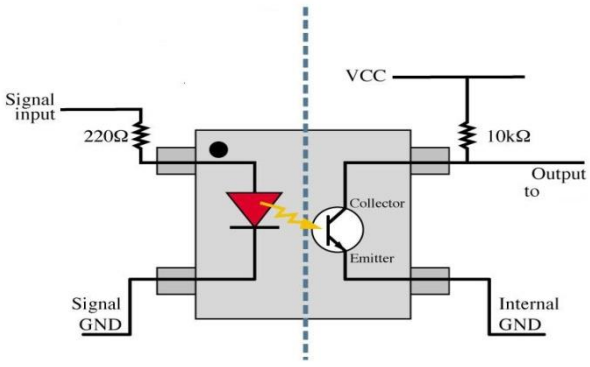

Fig 5.1 Opto coupler Isolation circuit diagram

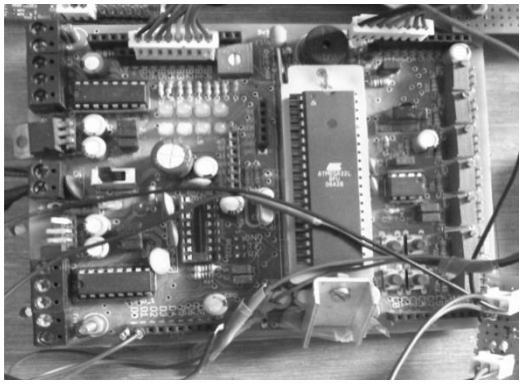

Fig 5.2 Our Microcontroller Circuit

6. RESULTS

The ROBOTIC arm with 6 DOF along with the robot was tested and its operation was noted. The visual studio 2010 provides an easy way to interface and control arm via serial port and its performance over internet is tested in different network connectivity. The payload capacity was determined and it was found as 4kgs. It was subjected to temperatures between 16 and 45 degree Celsius and there was no deterioration in performance measured [4]. The circuits in the ARM were affected by electromagnetic radiation due to the electronic devices and surroundings which caused significant errors which deteriorated the performance. This problem was solved by using shielded twisted pair cables to protect the signals from being affected by electromagnetic interference (EMI). Further all electronic devices were shielded to avoid being affected by EMI.

6.1. *Block Diagram:*

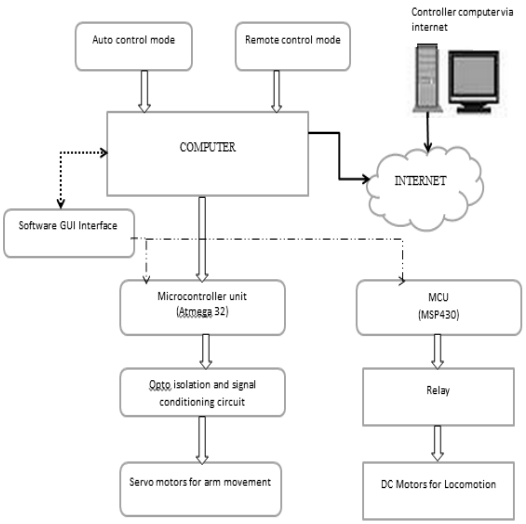

Fig 6.1 Basic function Block Diagram

The block diagram explains the complete system. The bi-directional lines indicate bi-directional data flow and control. The single arrow indicates a master to slave relationship. Based on the user selection the remote control or Automatic control of the ARM would operate accordingly. The Opto isolation circuit is used to shield the MC (Atmega32) from reverse current.

6.2 *Picture of Our Robotic Arm Model*

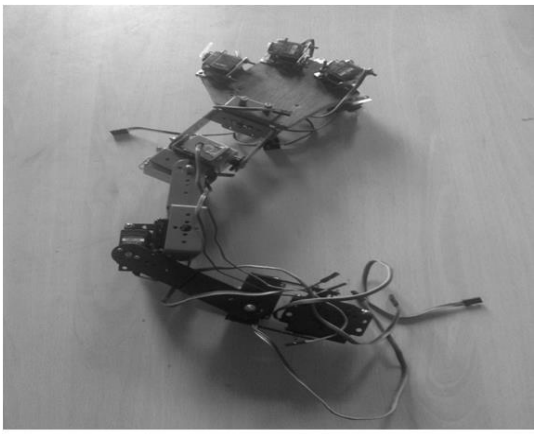

Fig 6.2 our servo motor assembled ROBOT ARM

7. CONCLUSION

The paper presents a study of the design of a 6 degree of freedom robot ARM in order to replicate a human arm. It lists out the methods used for data communication over internet between arm and remote user and precautions taken to avoid errors. Future work is concentrated on (a) building high torque-servos to reduce the size of arm and to increase the Pay load capacity up to 10Kgs or higher (b) build multiple ROBOT Arm's and to make them coordinate with each other through IP based protocol and (c) incorporate fuzzy logic and artificial neural network model to constructed ROBOT.

REFERNCES

[1] Z.W. LUO, M. ONISHI, T.ODASHIMA, K. OYAMA, F. ASANO, S. HOSOE, "INTEGRATION OF PC-BASED 3D IMMERSION TECHNOLOGY FOR BIO-MIMETIC STUDY OF HUMAN INTERATIVE ROBOTICS". INTERNATIONAL CONFERENCE ON ROBOTICS, INTELLIGENT SYSTEMS AND SIGNAL PROCESSING CHANGSHA, CHINA, OCTOBER 2003. .

[2] W.I. CLEMENT, MEMBER, IEEE, AND K.A. KNOWLES, "AN INSTRUCTIONAL ROBOTICS AND MACHINE VISION LABORATORY". IEEE TRANSACTIONS ON EDUCATION, VOL. 37, NO. 1, FEBRUARY 1994.

[3] URMILA MESHRAM, PANKAJ BANDE, R.R. HARKARE,D.WARAMWAR "ROBOT ARM CONTROLLER USING FPGA" IEEE,IMPACT 2009, PP.8-11.
[4] J.M. LEE, B.S. PARK, Y.S. LEE, J.S. AHN, S.H. LEE, S.J. LIM, C.S. HAN, "THE DEVELOPMENT OF THE ROBOT MANIPULATOR FOR AN INTELLIGENT SERVICE ROBOT," INTERNATIONAL JOINT CONFERENCE ON SICE-ICASE, PP. 282-287, 2006.
[5] ATMEGA16 - 8-BIT AVR MICROCONTROLLER WITH 16K BYTES IN-SYSTEM PROGRAMMABLE FLASH - ATMEL CORPORATION DATASHEET.